

## **Avaliativa: Implementações do TAD Fila**

**Objetivo:** Capacitar ao aluno desenvolver as operações básicas de fila, utilizando as implementações estática/sequencial e dinâmica/encadeada simples.

### **Exercícios:**

1. Implementar, utilizando a alocação **estática**, o acesso **sequencial** e a estratégia que **usa um contador** de elementos, o TAD fila com até 10 *strings*, sendo que cada *string* deve ter, no máximo, 30 caracteres (desconsiderando o '\0'). Além das operações básicas (*cria\_fila*, *fila\_vazia*, *fila\_cheia*, *insere\_fim*, *remove\_ini*), o TAD também deve contemplar a operação *apaga\_fila*, que elimina a instância da fila passada como entrada, caso ela exista; *esvazia\_fila*, que retorna a fila de entrada para o estado de vazia; e *tamanho\_fila*. Também desenvolva um programa aplicativo que permita ao usuário criar uma fila, inserir e remover elementos, bem como apagá-la, esvaziá-la, imprimi-la e verificar seu tamanho, desde que ela já tenha sido criada.

*OBS: nesta estratégia não é necessário um campo para guardar a posição final da fila, uma vez que ela pode ser obtida a partir da posição inicial e do contador.*

*OBS2: a impressão de uma fila deve ser feita na aplicação, utilizando APENAS as operações básicas descritas acima. Apesar de menos eficiente, é o que o desenvolvedor da aplicação pode fazer no cenário descrito neste exercício, dado que NÃO existe a operação *get\_elem\_pos* (que contraria a política de acesso FIFO).*

2. Implementar, utilizando a alocação **dinâmica** e o acesso encadeado **simples**, o TAD fila de *produtos*. Cada *produto* deve guardar as seguintes informações: *código (inteiro)*, *descrição (com no máximo 10 caracteres)* e *preço (float)*. Esse TAD deve ter as mesmas operações (exceto *fila\_cheia*) e programa aplicativo da questão anterior.